Лабораторна робота №6

Інтеграція Nagios Plugins у Icinga 2 для розширеного моніторингу

Мета: Набути практичних навичок інтеграції та використання плагінів Nagios у системі Icinga 2 для розширеного моніторингу. Засвоїти перевірку доступності Windows- та Linux-хостів через відкриті TCP-порти за допомогою плагіна check_tcp, навчитися створювати власні користувацькі команди та шаблони сервісів у Icinga Director, виконувати деплой конфігурацій, а також проаналізувати особливості та наслідки використання гібридної конфігурації об'єктів у середовищі моніторингу.

Інструменти: гіпервізор VirtualBox, модель комп'ютерної мережі.

Теоретичні відомості

У попередніх лабораторних роботах було створено віртуалізоване стендове середовище у VirtualBox, що складається з чотирьох хостів:

Serv-G-N-1 (Windows Server 2022) – контролер домену з ролями AD DS, DNS і DHCP, на якому встановлено Icinga 2 Agent для локального моніторингу ресурсів.

Serv-G-N-3 (Ubuntu Server 24.04) – сервер моніторингу з Icinga 2, веб-інтерфейсом Icinga Web 2, базою даних Icinga DB та Icinga Director для централізованого керування конфігураціями.

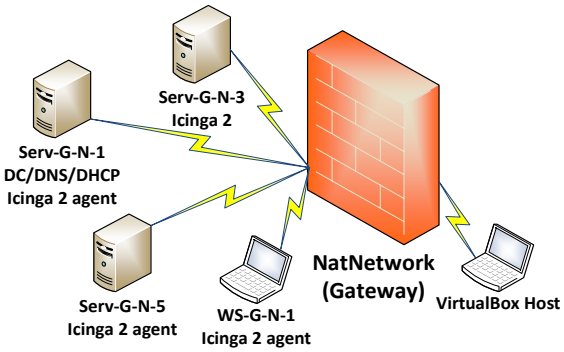

Рис. 6.1. Топологія мережі

WS-G-N-1 (Windows 10) – робоча станція, включена до внутрішнього домену, з встановленим Icinga 2 Agent; додана до системи моніторингу через Icinga Director.

Serv-G-N-5 (Ubuntu Server 24.04) – сервер з Icinga 2 Agent для локального моніторингу ресурсів.

Мережеве середовище забезпечує взаємодію між вузлами, а система Icinga 2 інтегрована з доменною інфраструктурою для подальшого моніторингу її елементів.

Перевірка доступності хостів і стану сервісів у Icinga 2

Перевіряємо доступність хостів і стану сервісів у Icinga 2. Входимо у веб-інтерфейс Icinga Web 2, обираємо меню Overview – Total Overview та Overview – Hosts.

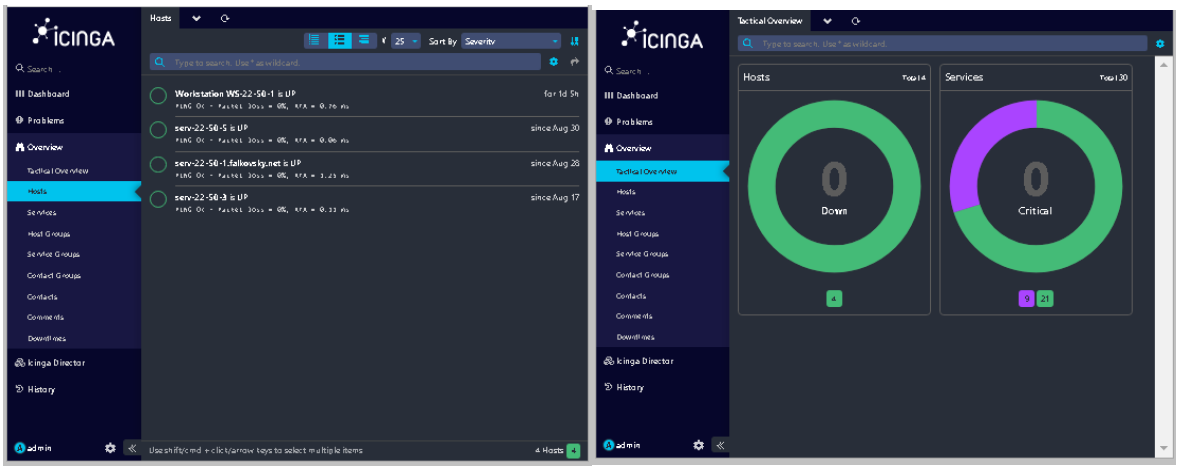

Рис. 6.2. Icinga Web 2. Перегляд меню Overview – Total Overview та Overview – Hosts

Перевіряємо, що всі хости (Serv-G-N-1, Serv-G-N-3, WS-G-N-1, Serv-G-N-5) мають статус UP. Базові сервіси на кожному хості не мають помилок чи стану Critical/Warning. Якщо будуть виявлені помилки, усуваємо їх і підтверджуємо нормальний стан хостів та сервісів. Відсутність помилок та Critical/Warning надає можливість для інтеграції Nagios Plugins.

Також перегляньте та ознайомтесь з інформацією у пунктах меню Overview – Services, Overview – ServiceGroups, Problems – Host Problems та Problems – Service Problems.

Перевірка доступності Windows- та Linux-хостів через відкриті порти.

Для перевірки доступності хостів через відкриті порти використаємо (Nagios Plugin check_tcp). У веб-інтерфейсі Icinga Web 2 відкриваємо меню Icinga Director – Commands – External Commands, де зберігаються команди, які використовують Nagios Plugins та перевіряємо присутність команди check_tcp.

Якщо команда відсутня, її можна встановити на сервері Icinga 2 через пакет ***monitoring-plugins-basic/standard***, після чого перевірити присутність виконуваного файлу за шляхом ***/usr/lib/nagios/plugins/check_tcp***. Встановлюємо пакет Nagios Plugins та перевіряємо наявність check_tcp:

sudo apt update
sudo apt install nagios-plugins-basic
ls /usr/lib/nagios/plugins/check_tcp

Для перевірки роботи команди у CLI використовуємо приклад:

/usr/lib/nagios/plugins/check_tcp -H <IP або FQDN хоста> -p <порт>

Перелік портів для моніторингу, що перевіряються на даному етапі для Windows-хостів:

ws-G-N-1.surname.net → порт 135 (RPC, доступ до служб Windows), 5665 (Icinga 2 Agent)
serv-G-N-1.surname.net → порт 135 (RPC), 5985 (WinRM), 53 (DNS), 5665 (Icinga 2 Agent)

Перелік портів для моніторингу, що перевіряються на даному етапі для Linux-хостів:

serv-G-N-3 → порт 22 (SSH), 5665 (Icinga 2 Agent)
serv-G-N-5 → порт 22 (SSH), 5665 (Icinga 2 Agent)

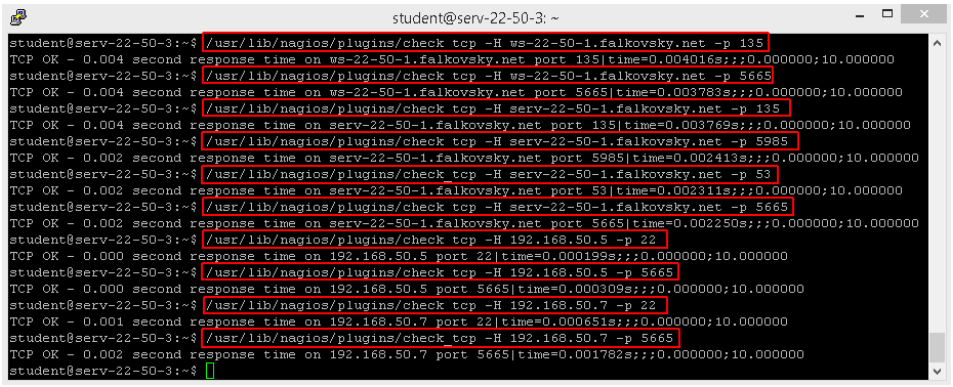

Рис. 6.3. Перевірка відкритості портів для моніторингу через CLI командою check_tcp.

Приклад команд та результатів перевірки, що підтверджують доступність портів наведено у додатку 1. У разі виникнення помилки типу CRITICAL або таймауту слід перевірити відкритість порту на хості та налаштування брандмауера. Ця перевірка дозволяє впевнитися, що критичні сервіси на Windows- та Linux-хостах доступні для подальшого моніторингу через Icinga 2 (рис.6.3).

Додамо перевірки портів у Icinga Director. Для цього створимо користувацьку команду перевірки.

У Director є два підменю:

- ✓ ***External Commands*** – перелік вбудованих/імпортованих команд, які Icinga підтягує з базових пакетів (nagios-plugins, monitoring-plugins тощо). Вони з'являються там автоматично після імпорту
- ✓ ***Commands*** – власний простір, де створюються й зберігаються кастомні команди.

У веб-інтерфейсі Icinga Web 2 переходимо у меню Icinga Director – Commands – Commands і створюємо вручну нову команду check_tcp_custom по кнопці Add. Заповнюємо поля наступним чином:

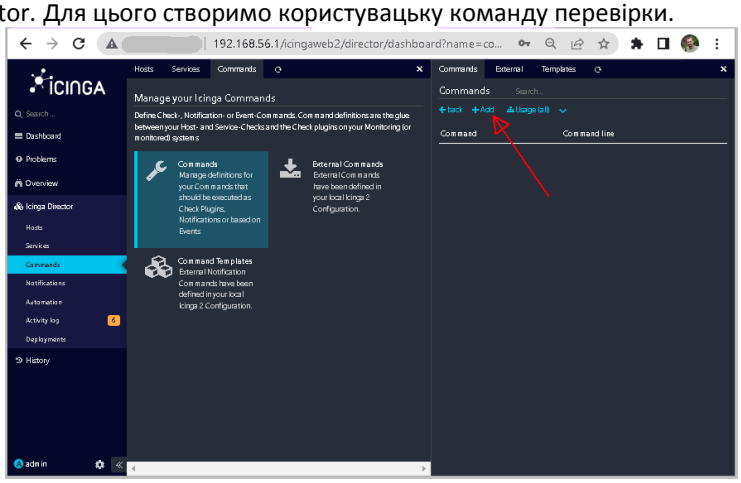

Рис. 6.4. Icinga Director – Commands – Commands – Add

- Command type: Plugin Check Command
- Command name: check_tcp_custom
- Command: check_tcp
- Render as string: No
- Timeout: 30
- Disabled: No

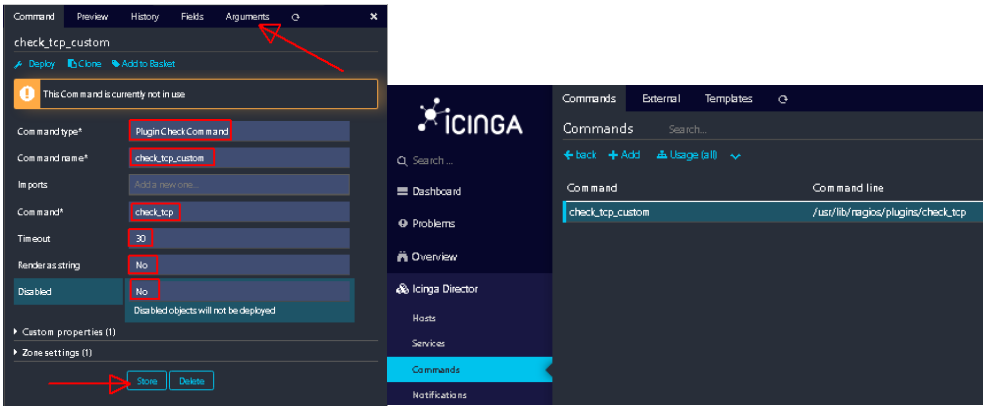

Рис. 6.5. Створення користувацької команди check_tcp_custom.

Коротко про заповнення аргументів: у полі -H підставляємо IP/FQDN хоста через рядок символів ***"hostaddress"***, а у полі -p — порт як змінну сервісу через рядок ***"port"***. Не вірні символи у цих рядках можуть порушити сприйняття формату аргументу в майбутньому. Прапорець ***Render as string = No*** гарантує, що команда буде збережена у форматі масиву (Array), а не рядка. Це обов'язково для того, щоб Icinga могла правильно обробляти аргументи з підстановкою змінних.

Заходимо у меню Arguments створеної команди ***check_tcp_custom*** та додаємо перший аргумент:

- Argument name: -H
- Description: Host address
- Value type: String
- Value: $host.address$
- Required: Yes

Зберігаємо перший аргумент через кнопки Add та Store та додаємо другий аргумент:

- Argument name: -p
- Description: TCP Port
- Value type: String
- Value: $port$
- Required: Yes

Зберігаємо зміни через Add, Store. Результатом має стати два налаштованих аргументи (рис.6.6.)

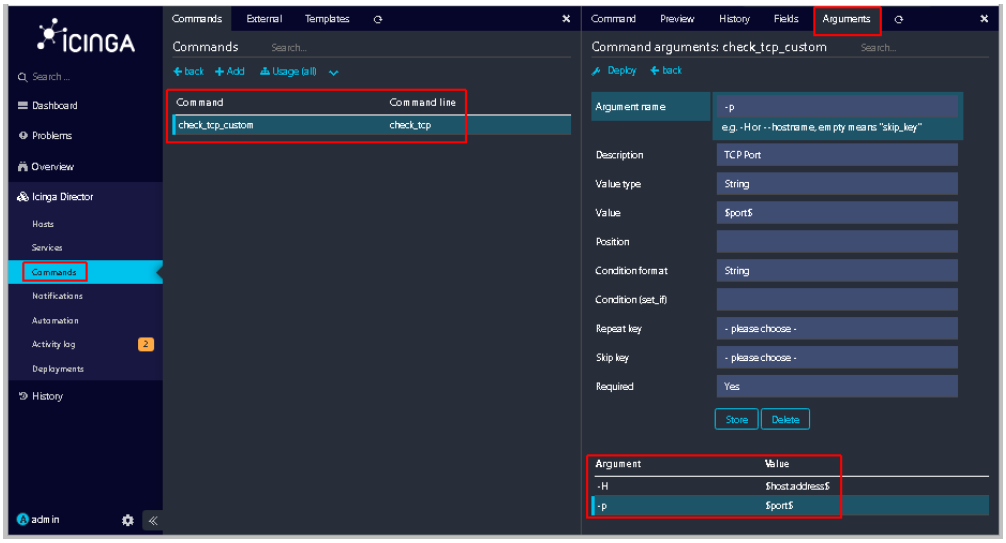

Рис. 6.6. Налаштування «аргументів» команди check_tcp_custom.

Маючи два налаштованих аргументи в команді (-H для адреси хоста та -p для порту), перемикаємось на закладку Fields і створюємо користувацьке поле тільки для аргументу -p.

Варто підкреслити, що в Director вже існує стандартна змінна ***$host.address$***, тому створювати додаткове поле для аргументу ***$host.address$*** немає потреби.

У закладці Fields створюємо поле для порту (рис.6.7):

- Field port(port)
- Description TCP port number that should be checked
- Mandatory Mandatory

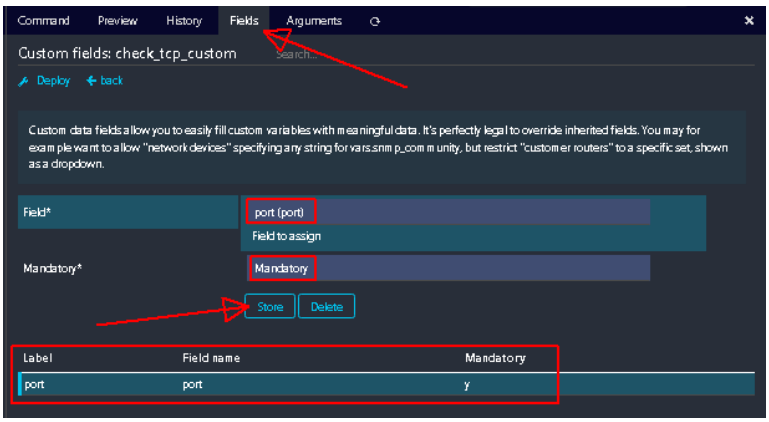

Рис. 6.7. Налаштування «полів» для «аргументу» port команди check_tcp_custom.

Створюємо шаблон сервісу tcp-port-check. Меню Icinga Director – Services – Service Templates – Add та заповнюємо поля наступним чином:

- Name: tcp-port-check
- Check command: check_tcp_custom (команда, створена на попередньому кроці)
- Check interval: 5m (1 раз на 5 хвилин)
- Retry interval: 30s
  (якщо перша перевірка впала — через скільки секунд пробувати ще раз)
- Max check attempts: 3 (скільки разів пробувати перед тим як оголосити CRITICAL)
- Check timeout: 10s
- Execute active checks: Yes
- Accept passive checks: No
- Send notifications: Yes
- Enable event handler: No
- Process performance data: No
- Enable flap detection: No
- Volatile: No

Інші поля (Notes, URL, картинки) можна не чіпати.

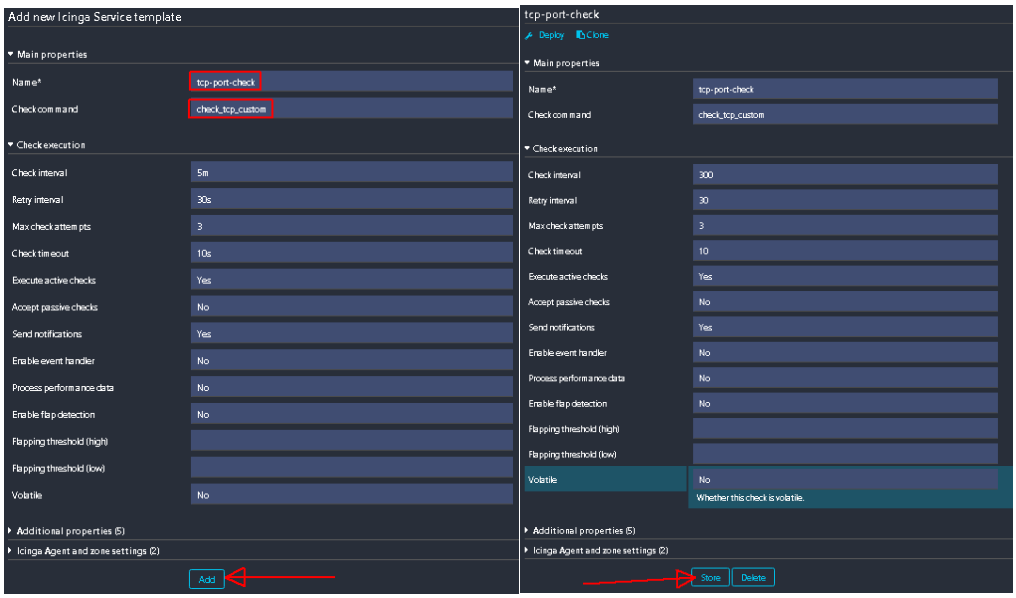

Рис. 6.8. Створення користувацького шаблону tcp-port-check.

Зберігаємо створений шаблон tcp-port-check. Шаблон очікує, що під час створення сервісу для конкретного хоста буде надано значення змінної port (номер TCP-порту, який треба перевіряти). Адреса хосту автоматично підтягується через стандартну змінну ***$host.address$***. Сам шаблон є "порожнім" — він лише визначає логіку перевірки. Працює він тільки тоді, коли ми створюємо конкретний сервіс і передаємо йому потрібні змінні.

Переходимо в Icinga Director – Services – Single Services та натискаємо Add та обираємо хост, для якого створюється перевірка (наприклад ws-22-50-1). Завантажиться форма «Add new Icinga Service», де у полі

Name необхідно вказати зрозумілу назву перевірки. Наприклад RPC tcp-135, WinRM tcp-5985 або HTTPS tcp-443. Назва не обов'язково має збігатися з ім'ям команди чи шаблону. Це чисто "людське" ім'я сервісу, яке потім відображатиметься в списку сервісів у вебінтерфейсі та дозволяє мати кілька сервісів на одному хості з різними портами. Приклад заповнення форми для перевірки RPC та Icinga Agent tcp-5665:

- Name: RPC tcp-135
- Imports: tcp-port-check
- Host: ws-22-50-1.falkovsky.net
- Disabled: No
- Check command: check_tcp_custom

Custom Properties

- port: 135

- Name: Icinga Agent tcp-5665
- Imports: tcp-port-check
- Host: ws-22-50-1.falkovsky.net
- Disabled: No
- Check command: check_tcp_custom

Custom Properties

- port: 5665

Таким чином IP/FQDN хосту береться з ***$host.address$*** а порт задається вручну у полі port. На рис. 6.9 показане додавання перевірок доступності портів 135 та 5665 для хосту ***ws-22-50-1.falkovsky.net***

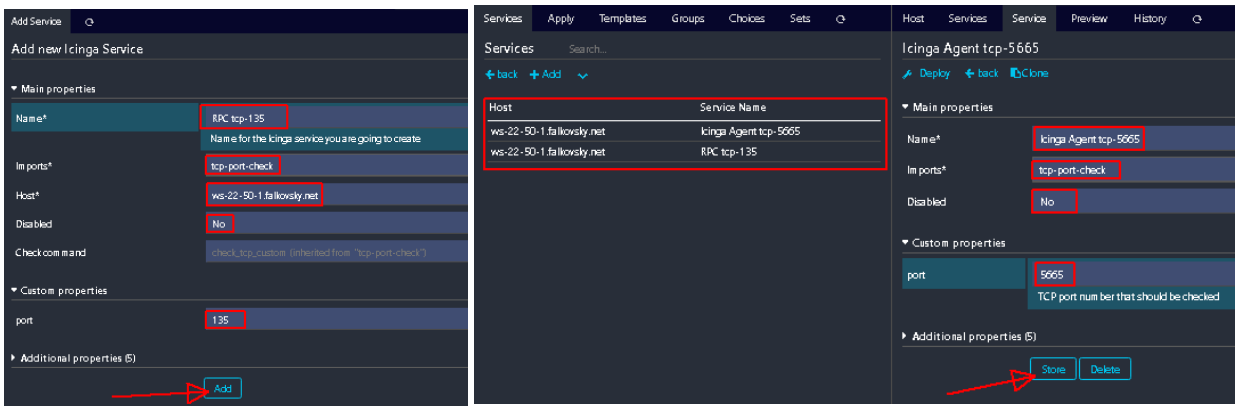

Рис. 6.9. Додавання двох сервісів для ws-22-50-1.falkovsky.net у Icinga Director

Деплой конфігурації доданих сервісів у Icinga Director

Перед застосуванням змін у системі моніторингу рекомендується створити резервний бекап віртуальної машини (snapshot) або скопіювати каталог конфігурацій Icinga2:

sudo cp -r /etc/icinga2 /etc/icinga2.backup.$(date +%F)

Це дозволить безпечно відкотитися у разі помилок під час деплою.

Для деплою через Director у веб-інтерфейсі переходимо у меню Deployments – Render Config. На сторінці Generated Config відобразиться список файлів і змін, що будуть застосовані. У блоці Actions, натискаємо Deploy (підпис може відрізнятися залежно від кількості змін та номеру «деплою»). Deploy застосовує всі накопичені зміни відразу для всієї конфігурації.

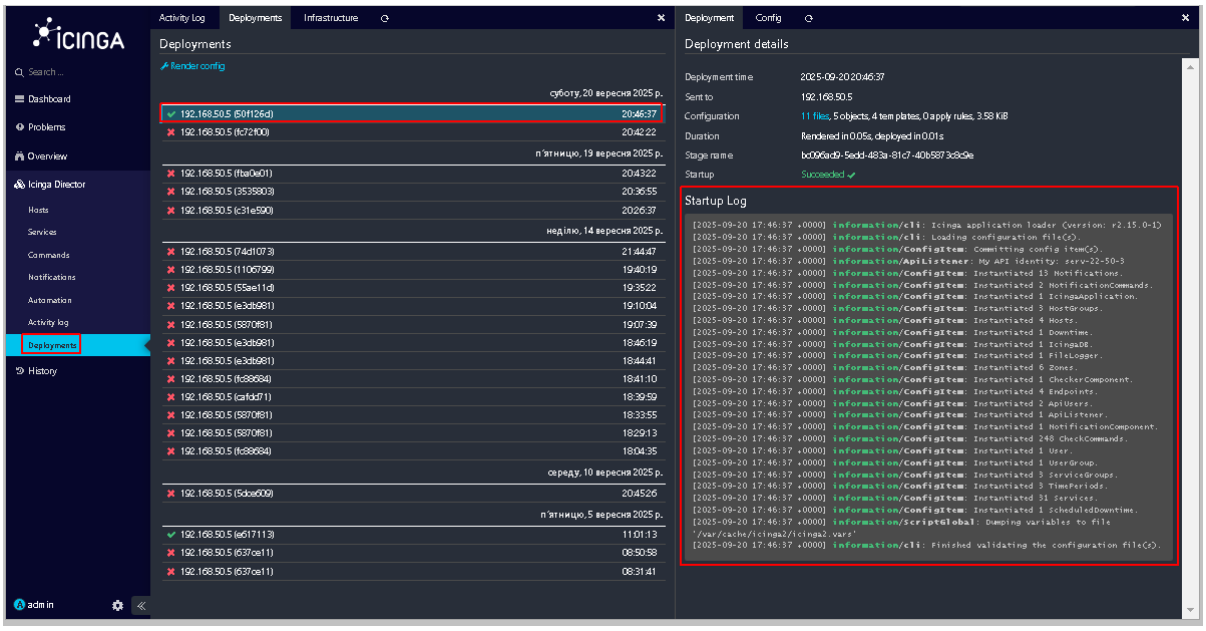

Рис. 6.10. Результат успішного «деплою» сервіса для ws-22-50-1.falkovsky.net у Icinga Director

При створенні нового сервісу у меню Icinga Director – Services – Single Services – Add у списку вибору хостів можна побачити лише один об’єкт — той, що був створений безпосередньо у Director. Інші хости, які були описані вручну у конфігураційних файлах (/etc/icinga2/zones.d/...), у цьому списку відсутні.

Ця ситуація виникла через те, що Icinga Director працює виключно зі своєю базою даних і не читає напряму файлів з zones.d. Тобто для Director «існують» лише ті хости, які були додані через його інтерфейс. У той же час Icinga Core бачить і моніторить усі хости незалежно від способу їхнього створення, тому у вебінтерфейсі моніторингу вони відображаються коректно.

Спробуємо додати у меню Icinga Director – Hosts – Hosts – Add сервер контролеру домену serv-22-50-1 що був доданий до системи моніторингу «вручну» через конфігураційні файли (рис.6.11).

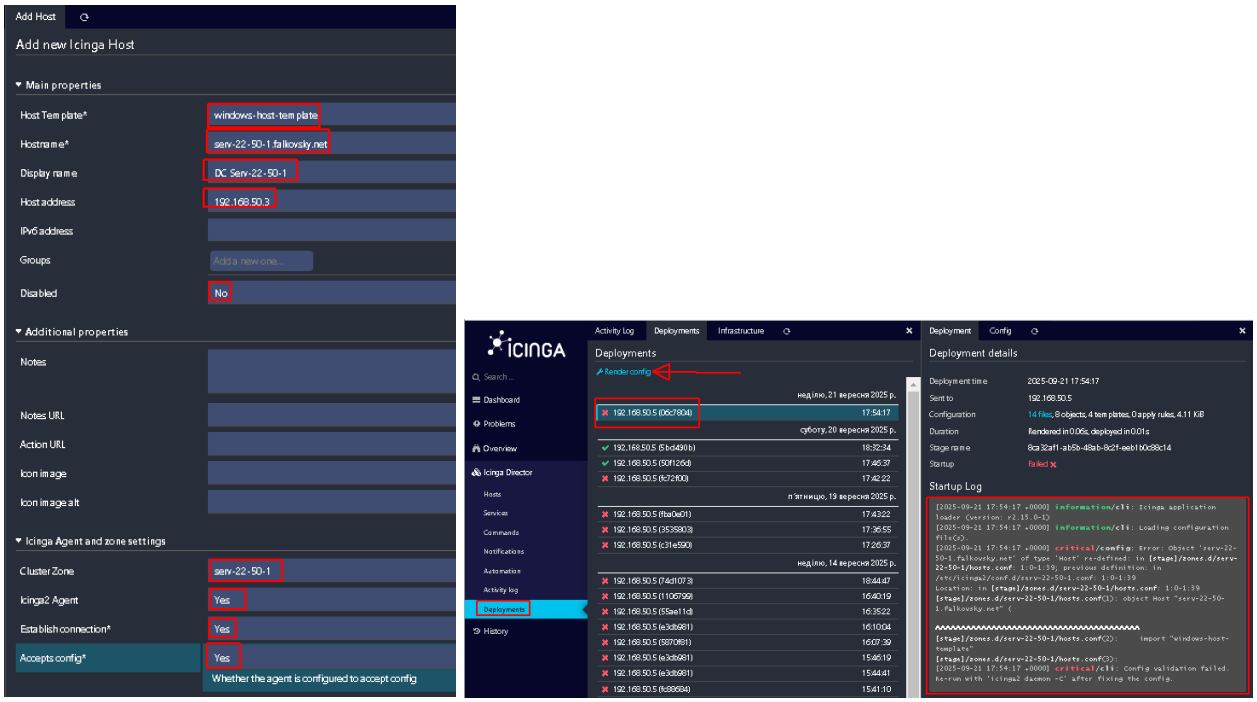

Рис. 6.11. Додавання та спроба деплою хосту serv-22-50-1.falkovsky.net у Icinga Director

Як і було передбачено, при деплої хосту ми отримуємо помилку, що двічі описано один і той самий об’єкт Host serv-22-50-1.falkovsky.net:

```
[2025-09-21 17:54:17 +0000] critical/config: Error: Object 'serv-22-50-1.falkovsky.net' of type 'Host' re-defined: in [stage]/zones.d/serv-22-50-1/hosts.conf: 1:0-1:39; previous definition: in /etc/icinga2/conf.d/serv-22-50-1.conf: 1:0-1:39
Location: in [stage]/zones.d/serv-22-50-1/hosts.conf: 1:0-1:39
[stage]/zones.d/serv-22-50-1/hosts.conf(1): object Host "serv-22-50-1.falkovsky.net" {
                                            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
[stage]/zones.d/serv-22-50-1/hosts.conf(2):     import "windows-host-template"
[stage]/zones.d/serv-22-50-1/hosts.conf(3):
[2025-09-21 17:54:17 +0000] critical/cli: Config validation failed. Re-run with 'icinga2 daemon -C' after fixing the config.
```

Найбільш коректним рішенням у такій ситуації було б повністю перевести всі хости до Director і працювати з єдиним джерелом конфігурації. Проте це потребує суттєвих трудозатрат на перенесення вже налаштованих хостів та сервісів. У нашому випадку це недоцільно, оскільки ми працюємо з лабораторним стендом, який використовується виключно для навчальних цілей і не експлуатується у продуктивному середовищі.

Приймаємо компромісне рішення — залишаємо гібридну конфігурацію: частина об’єктів описана вручну у конфігураційних файлах, а частина створюється через Director. Такий підхід у загальному випадку є недоліком (може призвести до плутанини та ускладнити адміністрування), але в рамках циклу лабораторних робіт він повністю задовольняє наші потреби і водночас дозволяє ознайомитися з обома методами керування конфігурацією.

Оскільки дублювання об’єкта Host serv-22-50-1.falkovsky.net призводить до помилки, необхідно видалити його запис із Director, залишивши лише початкове визначення у конфігураційних файлах. Переходимо у меню Hosts – Hosts – serv-22-50-1.falkovsky.net та видаляємо цей елемент з бази Director (рис.6.12).

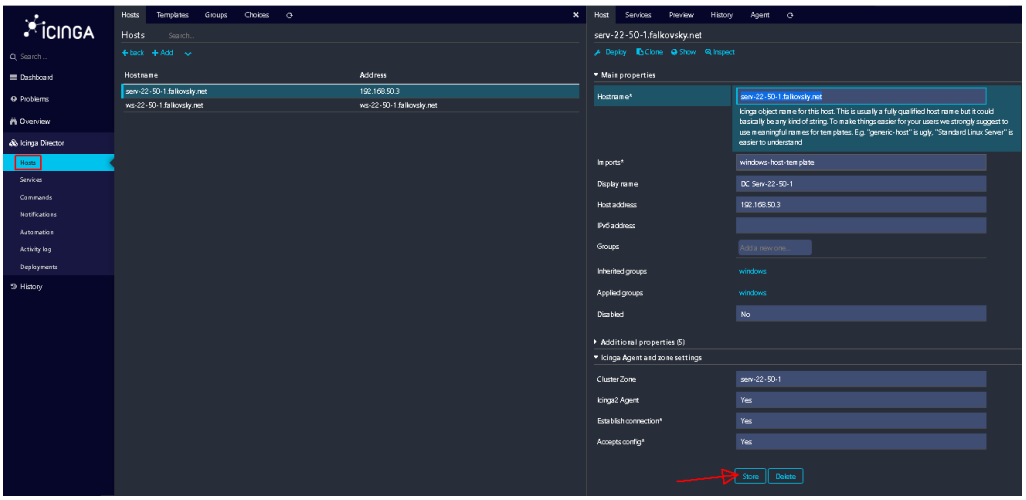

Рис. 6.12. Видалення хосту serv-22-50-1.falkovsky.net у Icinga Director

Завдання до лабораторної роботи

1. Встановити пакет nagios-plugins-basic та протестувати роботу плагіна check_tcp у CLI для вибраних портів Windows- та Linux-хостів.
2. Скласти перелік портів для моніторингу та перевірити їх доступність.
3. Створити у Director універсальний сервіс check_tcp_custom і налаштувати перевірку мінімум двох портів для Windows-робочої станції.
4. Виконати Config Deployment та перевірити відображення сервісів у вебінтерфейсі.
5. Проаналізувати проблему дублювання хостів, обґрунтувати вибір гібридної конфігурації та видалити дублюючий запис у Director.

Додаток 1.

Приклад команд check_tcp, що підтверджують доступність портів

```
/usr/lib/nagios/plugins/check_tcp -H ws-22-50-1.falkovsky.net -p 135
TCP OK - 0.004 second response time on ws-22-50-1.falkovsky.net port 135|time=0.004016s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H ws-22-50-1.falkovsky.net -p 5665
TCP OK - 0.004 second response time on ws-22-50-1.falkovsky.net port 5665|time=0.003783s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H serv-22-50-1.falkovsky.net -p 135
TCP OK - 0.004 second response time on serv-22-50-1.falkovsky.net port 135|time=0.003769s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H serv-22-50-1.falkovsky.net -p 5985
TCP OK - 0.002 second response time on serv-22-50-1.falkovsky.net port 5985|time=0.002413s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H serv-22-50-1.falkovsky.net -p 53
TCP OK - 0.002 second response time on serv-22-50-1.falkovsky.net port 53|time=0.002311s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H serv-22-50-1.falkovsky.net -p 5665
TCP OK - 0.002 second response time on serv-22-50-1.falkovsky.net port 5665|time=0.002250s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H 192.168.50.5 -p 22
TCP OK - 0.000 second response time on 192.168.50.5 port 22|time=0.000199s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H 192.168.50.5 -p 5665
TCP OK - 0.000 second response time on 192.168.50.5 port 5665|time=0.000309s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H 192.168.50.7 -p 22
TCP OK - 0.001 second response time on 192.168.50.7 port 22|time=0.000651s;;;0.000000;10.000000
/usr/lib/nagios/plugins/check_tcp -H 192.168.50.7 -p 5665
TCP OK - 0.002 second response time on 192.168.50.7 port 5665|time=0.001782s;;;0.000000;10.000000
```

Корисні посилання

- Icinga Director. Getting started.

https://icinga.com/docs/icinga-director/latest/doc/04-Getting-started/

- Icinga Director Setup Walktrough.

https://icinga.com/docs/get-started/latest/doc/12-director/

- First steps with Icinga Director.

https://community.icinga.com/t/first-steps-with-icinga-director/336